MicroServices with Devops Example

In this post I show you How to create Micro service application and Manage it through Dev-ops.

***To understand MicroServices please look at my blog post*** <http://javaonfly.blogspot.in/2016/07/microservices-basics.html>

***To understand Devops please look at my blog post***

<http://javaonfly.blogspot.in/2016/07/understanding-basics-of-devops.html>

Tools are Used to implement Microservice

1. Eclipse as Editor
2. Java as Language
3. Using REST Service to communicate between Microservices
4. Maven as build tool
5. Tomcat as Web server.

Tool are Used to Manage Microservices using Devops

1. Jenkins as CI (Continous integration server).
2. GitHub as SCM(Source control Management).

My intent is creating two Micro services

1. An Order Service
2. A Billing service

If client order for a product, **Order Service** project will communicate with **Billing service project** by **REST API**. **Billing Service** returns **Product price** if exists in the Product Catalog Otherwise It return a Message **Product not found**.

**Order service** accept this response and generate a new **response** to Client.

To Manage these **Microservice,** I using **Devops**

So that **Two can be deployed independently**, and Two projects are hosted in

**GitHub**. If someone do a change the code and **push** it in **Github** a build has been triggered by **Jenkins** and Once build is completed, **Jenkins** install the **artifact** that is the **war** file in **Local Repo** (maven Local REPO) then deploy it to **Tomcat Server**.

**No Manual intervention is needed.**

**I have made it very simple but in real time there will be much more complexity than two simple services and we need Monitoring tool for health check for different projects which is absent in this example.**

**I have used Java for Two services but real time one could be in java another in PHP or Node JS any language.**

**Installation:**

**Java:** Install java from

<https://java.com/en/download/help/index_installing.xml>

Set JAVA\_HOME in environment variables to java installation path.

**Eclipse** : <https://eclipse.org/downloads/>

**Tomcat** : <https://tomcat.apache.org/download-60.cgi>

Download Tomcat 6 Zip Version and extract it in your local drive that will your Tomcat installation directory.

**Maven:**

Download maven from <https://maven.apache.org/download.cgi>

Extract the zip in your local file Set M2\_Home as maven Installation path in your local directory.

**GIT**: Install GIt from <https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Go to Git Command prompt after installation

Change Directory to C: Using Cd ../../

Now apply command (Need Internet connection)

**git clone** [**https://github.com/shami83/MicroOrderService.git**](https://github.com/shami83/MicroOrderService.git)

**after completing**

**again hit**

**git clone** [**https://github.com/shami83/BillingService.git**](https://github.com/shami83/BillingService.git)

**Import two project in to Eclipse by import from local option.**

**If you want to create your new repositories, please go to Jenkins setup section and follow the step.**

**I prefer you create your new repositories to work independently. Unless if you clone it you will be dependent on me. If I gave you permission, then only you can push change on my project.**

**Jenkins :**

<http://mirrors.jenkins-ci.org/war-stable/> (Take stable release. I use 1.65). Download war and deploy it to <Tomcat\_installation\_path>/webapps. In tomcat console when a message shows

**“Jenkins full up and running” go to browser hit**

[**http://localhost:8080/jenkins**](http://localhost:8080/jenkins)

**First thing you need to require add plugins in to your local Jenkins Server**

So Follow the steps

Go to Manage Jenkins from left panel

Go to manage Plugin

In Available Tab Search with “git” and check gitplugin and github plugin

Hit install without restart button

Wait until installation finished.

**Go to configure system from manage plugin**

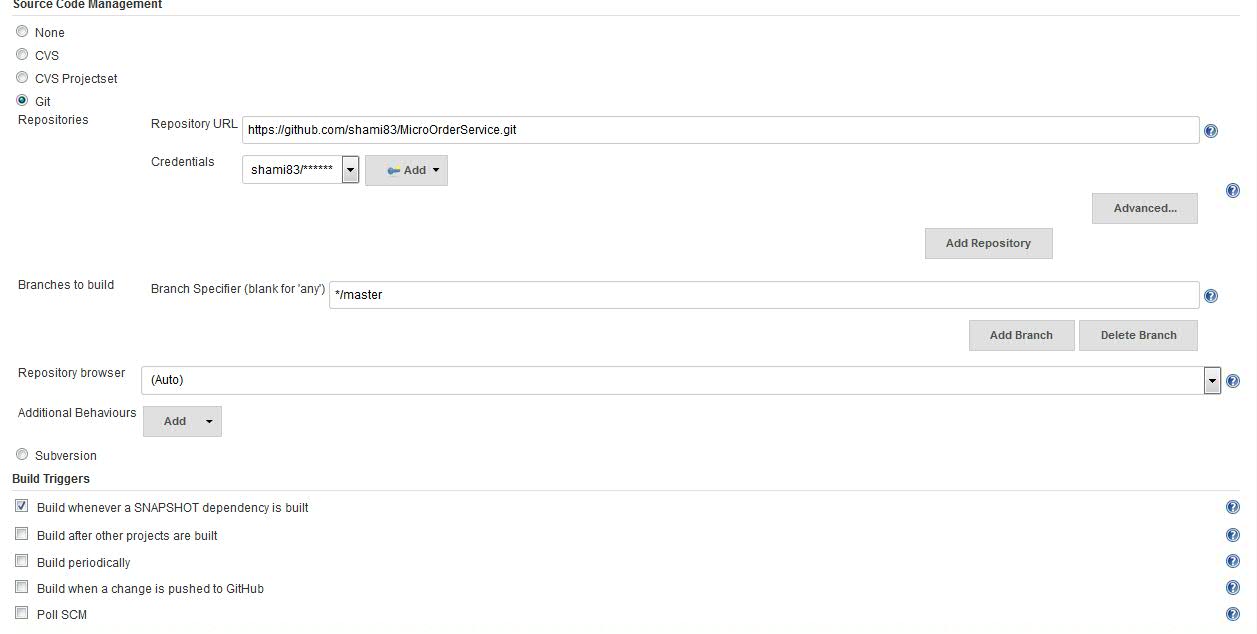
****

Create an account in GitHub and then add your username and your email Id.

Then Go to new Item create a Maven project.

Create a Job Name as “BuildandDeploy MicroOrderService”

Please see the following Screen shot

****

In the Source code management section set your Git repositart URL

If you download source code from Git by download Zip and create your new repository and push the code

**Steps are**

**After Download project from**

[**https://github.com/shami83/BillingService**](https://github.com/shami83/BillingService)

**and**

[**https://github.com/shami83/MicroOrderService**](https://github.com/shami83/MicroOrderService)

**using “ CloneorDownload” button on GitHub**

Extract Zip files

Create a project on Github using **“+”** button (Create new)

You will get a url like [**https://github.com/shami83/<projectname>**](https://github.com/shami83/%3cprojectname%3e)**.git**

**Execute following commands for two projects in your local computer**

Go to Git cmd

Go to the **homefolder of the Extract project**

1. **git init**
2. **git add .**
3. **git commit -m “demo”**
4. **git push origin master**

**Please do the steps for two projects that will create two repositories in Git (require internet connection).**

**In jenkins on repository url put the url**

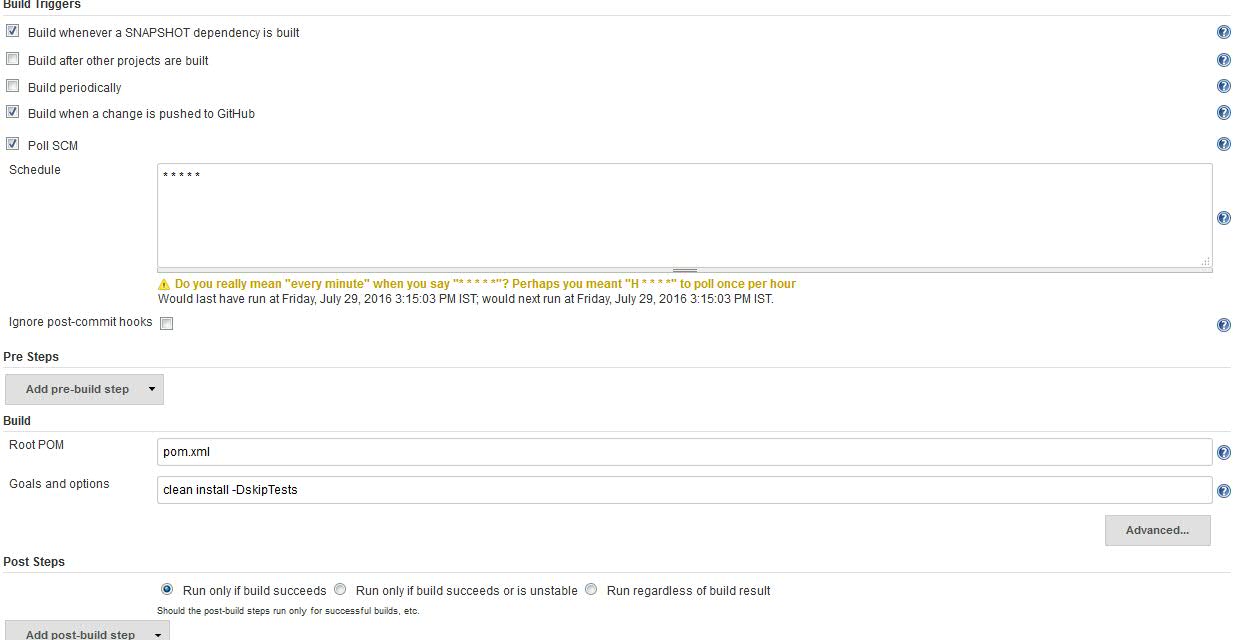
[**https://github.com/shami83/<projectname>.git**](https://github.com/shami83/%3cprojectname%3e.git)

**If you clone it then you need my permission to push code to my stream, please send me a mail the at**

[**mitrashamik@gmail.com**](mailto:mitrashamik@gmail.com) **or add a comment in blog.**

Then put your credentials in credential section your username and password of Git repository.

**Follow next step**

****

**In Build triggers section Check ” BuildWhen a change pushed to Git hub”**

**And poll SCM**

**Put value as \* \* \* \* \* in schedule text-area**

This will detect if any changes pushed on GitHub repository. If so this Jenkins target will run

Now Go to Build section

Set Root POM as pom.xml as you can see a workspace folder create under your Jenkins target where Jenkins pull the codebase from Git server and build it locally.

Set Goals and Option as **clean install.**

**Install step is requires as it maintains your artifacts in your local maven repository.**

**Last I want to deploy it on Tomcat server**

**To do that follow the following steps**

First go to Tomcat installation directory under conf folder

Open tomcat-users.xml in edit mode

Add following lines under <tomcat-users> tag.

**<role rolename="tomcat"/>**

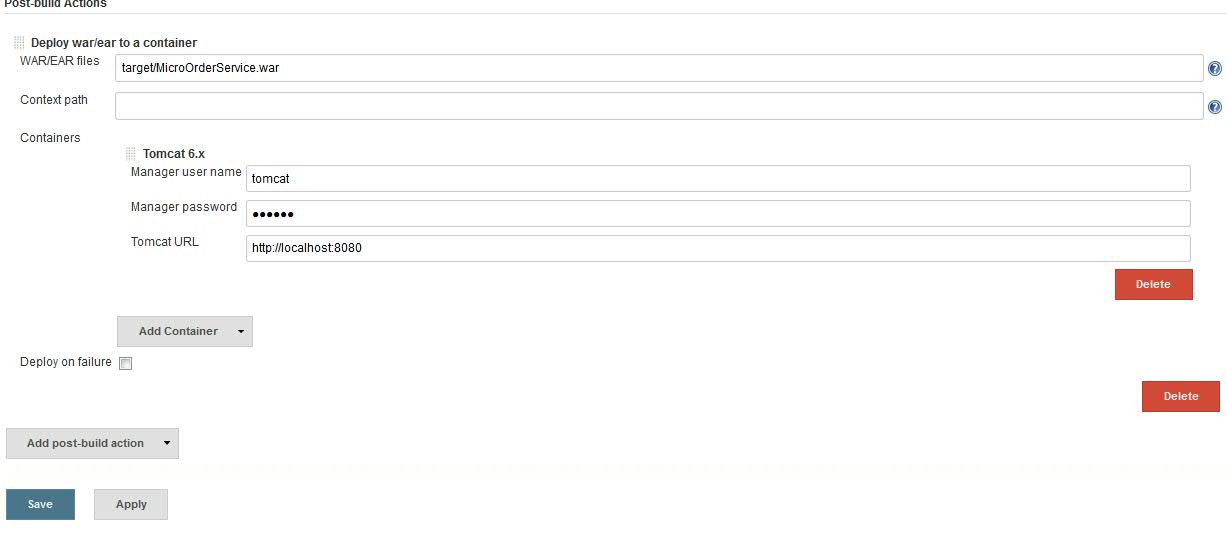
**<role rolename="role1"/>**

**<user username="tomcat" password="tomcat" roles="admin,manager-script,manager-gui"/>**

**<user username="both" password="tomcat" roles="tomcat,role1"/>**

**<user username="role1" password="tomcat" roles="role1"/>**

**Then go to Jenkins job configure section**

****

In War/EAR files put target\MicroOrderService.war as maven generates artifacts in target directory.

After that Set user name as tomcat in Tomcat6.x section password is tomcat

Tomcat Url is <http://localhost:8080>

Click save button that will create MicroOrderService job.

To check hit **BuildNow** button and go to **console output**, you will see

Jenkins pull code from GitHub then build it using maven install the artifacts then deploy it in your Tomcat Server

Perform Same thing for Billingservice, create a new Job and configure that.

Now to Check just make a change in Source code

Push it to Git using Git Cmd

Steps

git add .

git commit -m ”change”

git push origin master

go to Jenkins dashboard you will see a build is initiated and deployed.